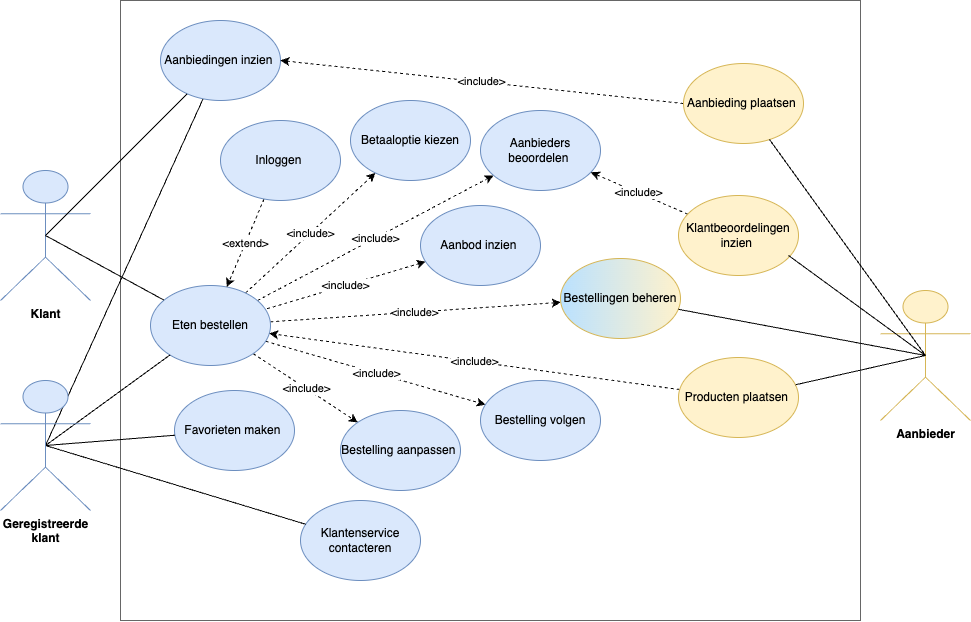
# Deelvraag: Wat zijn de verschillende onderdelen van de mobiele applicatie?

Voor het bepalen van de verschillende onderdelen die horden bij de te ontwikkelen mobiele applicatie moet eerst aandacht worden geschonken aan de functionaliteit die wordt verwacht van de applicatie. Hierbij wordt eerst middels usercases en userstories de gewenste functionaliteit geschetst. Hierna worden de verschillende functionele en niet-functionele aspecten beoordeeld. Tot slot wordt het C4 model voor de Homecooked applicatie geschetst. Aan de hand hiervan wordt er een goed beeld geschetst van de benodigde onderdelen van de applicatie.

# **Usecase Diagram**

Onderstaand is het usecase digram te zien van de homecooked mobiele applicatie. Hierbij wordt een onderscheid gemaakt tussen de klanten die eten bestellen en de aanbieders die eten en producten aanbieden via de applicatie.

De klanten zijn weergegeven in het blauw waarbij weer een onderscheid wordt gemaakt tussen eenmalige klanten en klanten die zijn geregistreerd. Klanten die zijn geregistreerd hebben meer opties zoals het beoordelen van aanbieders, favorieten aanmaken en klantenservicegegevens inzien tot de beschikking.



Figuur : Usecase Diagram Homecooked

# **User stories klanten**

Op basis van de bovenstaande usecases zijn de volgende userstories voorgekomen:

1. **Inzien aanbod:** Als gebruiker wil ik door de lijst van aanbieders kunnen bladeren om te zien wat zij aanbieden en wat de recensies van deze aanbieders en hun producten is, zodat ik een overwogen keuze kan maken.
2. **Bestellen van eten**: Als gebruiker wil ik voedsel kunnen bestellen van de aanbieders via de mobiele applicatie, zodat ik snel en eenvoudig het gewenste voedsel kan ontvangen of ophalen.
3. **Bestelling aanpassen:** Als gebruiker wil ik mijn bestellingen kunnen aanpassen zoals de samenstelling van het gerecht, zodat ik de smaak en pittigheid kan aanpassen aan mijn eigen wensen.
4. **Betaalopties kiezen:** Als gebruiker wil ik bij het bestellen uit verschillende betaalopties kunnen kiezen, zodat ik altijd de goede betaaloptie beschikbaar heb.
5. **Bestelling volgen:** Als klant wil ik mijn bestelling in real-time kunnen volgen, zodat ik weet wanner mijn bestelling zal aankomen en ik mij hierop kan instellen.
6. **Inloggen als klant:** Als frequente gebruiker wil ik kunnen inloggen met mijn email of met een sociale media-account, zodat ik mijn opgeslagen restaurants/aanbieders, bestelhistorie, betaalinformatie en persoonlijke gegevens kan inzien en wijzigen.
7. **Aanbieders beoordelen:** Als klant wil ik de aanbieders en hun gerechten individueel kunnen beoordelen, zodat ik mijn ervaring kan delen met andere gebruikers en de aanbieder, die vervolgens een overwegen keuze kunnen maken.
8. **Favorieten aanmaken:** Als vaste klant wil ik mijn favoriete restaurants en gerechten combinatie kunnen opslaan, zodat ik in de toekomst sneller mijn bestelling kan plaatsen.
9. **Aanbiedingen inzien:** Als gebruiker wil ik de lopende aanbiedingen kunnen inzien, zodat ik mijn keuze aan de hand van de aanbiedingen kan aanpassen en geld bespaar.
10. **Klantenservice contacteren:** Als gebruiker wil ik contact kunnen opnemen met de klantenservice, zodat ik hulp ontvang bij eventuele vragen of misverstanden.

# **User stories aanbieders**

1. **Producten plaatsen:** Als gebruiker wil ik aanpassingen kunnen maken via de app voor de beschrijvingen, de prijzen, de afbeeldingen, het assortiment en de informatie van mijn restaurant.
2. **Bestellingen beheren:** Als gebruiker wil ik de bestellingen ontvangen en kunnen beheren, zodat ik mijn klanten goed kan bedienen.
3. **Klantbeoordelingen inzien:** Als gebruiker wil ik mijn klantbeoordelingen kunnen beheren, zodat ik kan inzien wat de beoordelingen zijn en eventueel kan reageren op de beoordelingen.
4. **Aanbiedingen plaatsen:** Als gebruiker wil ik aanbiedingen kunnen plaatsen en beheren, zodat ik gebruikers kan aansporen om een bestelling te plaatsen.

# **Vereisten**

**Functionele eisen:**

Inzien van aanbod

* De gebruiker moet in een lijst alle beschikbare aanbieders kunnen inzien. Op deze lijst moet de naam, het aantal reviews, de locatie, de afstand, de soort keuken en een afbeelding van de aanbieder te zien zijn.
* De Gebruiker moet de mogelijk hebben om te zoeken naar naam tussen de aanbieders.
* De gebruiker moet door op de aanbieder te drukken een lijst van alle mogelijke producten die deze aanbieder heeft kunnen inzien. In de lijst is de naam en de prijs van de producten zichtbaar.

Bestellen van eten

* De gebruiker moet de verschillende gerechten kunnen inzien en toevoegen aan zijn winkelwagen.
* De gebruiker moet zijn winkelwagen kunnen inzien, hierop moeten de producten het aantal, de prijs en de totaalprijs van de producten zichtbaar zijn.
* De gebruiker moet in zijn winkelwagen het aantal producten kunnen wijzigen.

Inloggen

* De gebruiker moet bij het opstarten van de applicatie de keuzemogelijkheid kunnen hebben om in te loggen of een account aan te maken middels zijn email.
* De gebruiker moet middels zijn email en een wachtwoord kunnen inloggen in de applicatie.
* De gebruiker moet middels de biometrische mogelijkheden die zijn apparaat ondersteund kunnen inloggen.

Betalen

* De gebruiker moet bij het bestellen van de gerechten de keuzemogelijkheid hebben om een betaaloptie te kiezen. (PayPal, Klarna, IDEAL, Creditcard).
* De Gebruikers moeten middels een knop/vink de mogelijkheid hebben om de betaalgegevens op te slaan voor een volgend gebruik.

Bestellen

* De gebruiker moet de producten uit zijn winkelwagen kunnen bestellen en daarbij een keuze kunnen maken tussen bezorgen en afhalen. Tevens moet er bij iedere keuze een schatting van de af te halen tijd dan wel de bezorgtijd vermeld staan.
* Na het bestellen moet de gebruiker in het overzicht van de bestellingen de bestelling kunnen volgen, waarbij de afstand of het aantal geschatte minuten worden weergegeven.
* De gebruiker kunnen in het besteloverzicht de locatie van de bezorger volgen.
* De gebruiker moet na het plaatsen van een bestelling de vraag krijgen of hij of zij een review wilt plaatsen.

Account

* De gebruiker moet zijn accountgegevens op een accountpagina kunnen inzien. Hier moeten de avatar, de persoonsgegevens, betaalgegevens en adresgegeven zichtbaar zijn.
* De gebruiker moet in staat zijn al zijn gegevens te kunnen wijzigen in het accountscherm.
* De gebruiker moet kunnen uitloggen, zijn biometrische inlogmethode aanzetten en zijn account kunnen verwijderen.

Beoordelen

* De gebruiker moet in de lijst van aanbieders op reviews kunnen drukken om alle reviews van de aanbieders te kunnen inzien.
* De gebruikers moeten hun eigen beoordelingen kunnen inzien en kunnen verwijderen.

Overige

* De gebruikers moeten op de detailweergave van de aanbieders de actuele aanbiedingen kunnen inzien.
* De gebruikers moeten via de applicatie alle klantenservice gerelateerde gegevens kunnen inzien.
* Gebruikers moeten aanbieders en gerechten als favorieten kunnen aanmerken en deze in het favorieten scherm kunnen selecteren. Ook moeten de favorieten zichtbaar zijn in de lijst van aanbieders.

Aanbieders

* Aanbieders kunnen aanbiedingen plaatsen op hun assortiment.
* Aanbieders kunnen hun lopende aanbiedingen beheren.
* Aanbieders kunnen gebruikerservaringen inzien en beheren.
* Aanbieders kunnen reageren op gebruikerservaringen.
* Aanbieders kunnen de geplaatste bestellingen beheren.
* Aanbieders kunnen hun assortiment van producten plaatsen.
* Aanbieders kunnen de soort aanpassingen aangeven per product.

**Niet-functionele eisen:**

Voor de niet functionele eisen is een onderscheid gemaakt tussen de benodigde eisen voor de Homecooked applicatie en eisen die ook voor de POC’s zal gelden. Hierbij wordt eerst een volledige lijst met eisen weergegeven en hierna de lijst die specifiek voor de POC’s van toepassing is. Dit verschil is te verklaren omdat de POC’s niet alle functionaliteit van de uieindelijke applicatie zullen bevatten.

* De app moet de gebruiker token voor het valideren van de gebruiker om de veilig native oplossing van het systeem opslaan. Deze token moet een geldigheid hebben van 6 maanden.
* De lijst van aanbieders moet binnen 1 seconden doch uiterlijk binnen 3 seconden worden getoond aan de gebruikers uitgaand van een 4G verbindingssnelheid, indien niet binnen 1 seconden de lijst verschijnt moet de gebruiker een laadscherm te zien krijgen.
* Bij het scrollen tussen aanbieders mag het aantal frames niet zakken onder de 50FPS. Uitgaand van een apparaat dat in de middenklasse valt. (vanaf 550 euro, 2023 norm).
* De gebruiker moet op het hoofdscherm kunnen schakelen tussen accountgegevens, aanbieders en winkelwagen.
* De account- en betaalgegevens van de gebruiker moeten alleen versleuteld middels een hashing algoritme worden opgeslagen. Dit hashing algoritme moet voldoen aan de ‘industry standard’.
* De status van een bestelling mag niet ouder zijn dan 5 minuten.
* De klantenservicegegevens mogen niet ouder zijn dan 2 dagen.
* De app moet 10000 gebruikers tegelijkertijd aan kunnen zonder dat er vertraging bij de gebruikers zichtbaar is.
* Bij het wisselen van applicaties moeten de app onthouden waar gebruikers zijn gebleven.
* De app moet compatibel zijn met toestellen waarbij het besturingssysteem niet ouder is dan 6 jaar.
* De app mag niet crashen, en indien dit wel gebeurt mag de gebruiker hier geen hinder van ervaren door een vastlopende telefoon.
* De app moet een consistente gebruikerservaring bieden op de verschillende appraten. Hierbij moeten de themakleuren altijd gelijk blijven.
* De applicatie moet voldoen aan de WCAG (EN 301 549) richtlijnen.
* De applicatie moet ondersteuning bieden voor Engels, Duits, Frans en Nederlands.
* De applicatie moet compatible zijn voor de volgende schermformaten (3.5inch tot 6.5inch)
* De applicatie moet voldoen aan de AVG gegevens bescherming.
* Zowel portret- als landschapsmodus moeten compatible zijn en mogen geen afwijkingen vertonen.
* De app moet relevante pushmeldingen kunnen versturen. Hierbij mag het aantal meldingen niet hoger zijn dan 1 per 5 minuut en indien het reclame gerelateerde meldingen betreft dan zijn die beperkt tot 1 per dag.

Voor het onderzoek worden een tweetal POC’s gemaakt. De niet-functionele eisen zijn aangepast aan de functionaliteit van de POC’s. Hiervoor zijn aspecten meegenomen die daadwerkelijk kunnen worden getest met de POC’s. De volgende niet-functionele eisen zijn van toepassing op de POC’s

* De lijst van aanbieders moet binnen 1 seconden doch uiterlijk binnen 3 seconden worden getoond aan de gebruikers uitgaand van een 4G verbindingssnelheid, indien niet binnen 1 seconden de lijst verschijnt moet de gebruiker een laadscherm te zien krijgen.
* De app mag niet crashen, en indien dit wel gebeurt mag de gebruiker hier geen hinder van ervaren door een vastlopende telefoon.
* De app moet compatibel zijn met toestellen waarbij het besturingssysteem niet ouder is dan 6 jaar.
* De gebruiker moet op het hoofdscherm kunnen schakelen tussen accountgegevens, aanbieders en winkelwagen. (verplaatsen naar functionele)
* Bij het wisselen van applicaties moeten de app onthouden waar gebruikers zijn gebleven.
* Zowel portret- als landschapsmodus moeten compatible zijn en mogen geen afwijkingen vertonen.
* Bij het scrollen tussen aanbieders mag het aantal frames niet zakken onder de 50FPS. Uitgaand van een apparaat dat in de middenklasse valt. (vanaf 550 euro, 2023 norm).

## **C4 Model**

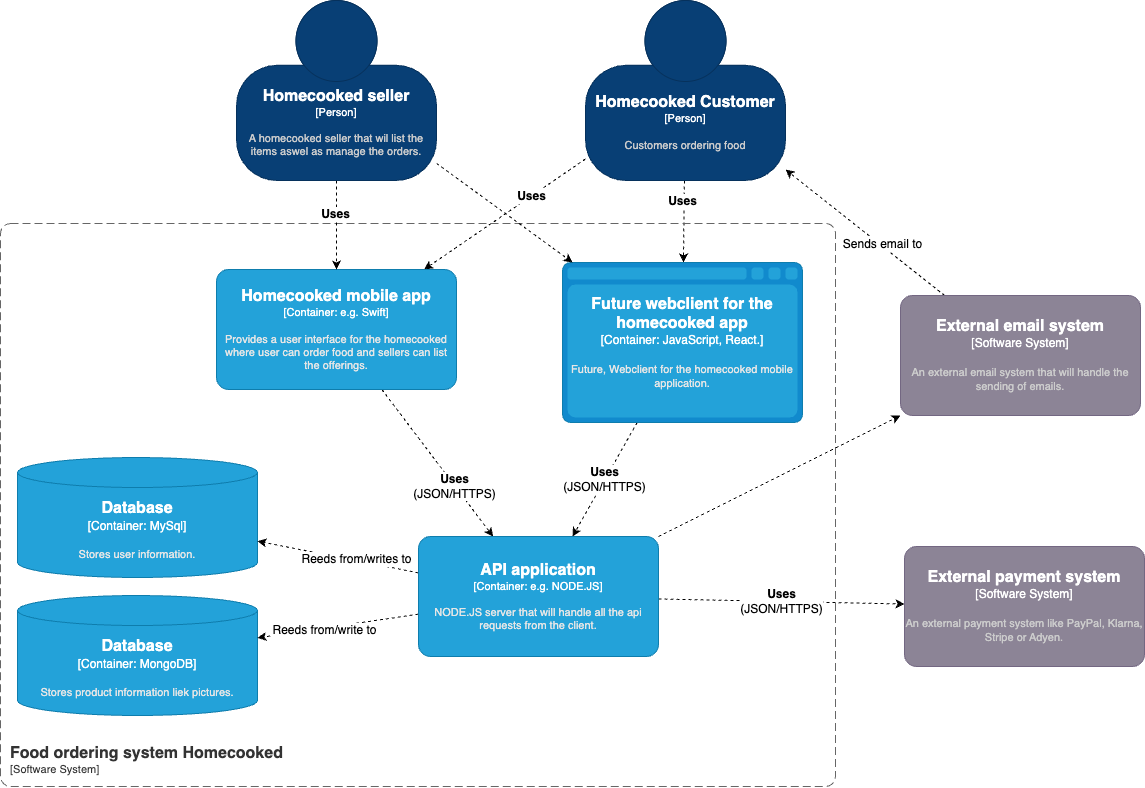
Onderstaand is de Homecooked architectuur visueel weergegeven middels het C4 model. Hierbij is een onderscheid gemaakt tussen de klanten en de aanbieders. Tevens is een webclient toegevoegd als toekomstige project.

In het C4 model is te zien hoe de mobiele client applicatie middels een API-server communiceert met de twee databases. Voor de API-server is gekozen voor Express.js dat draait op NODE.js. Door zijn minimalisme en flexibiliteit is het zeer geschikt voor een startup. Frameworks zoals Kraken, Feathers, NestJs, Sails en Locomotive maken allemaal gebruik van Express.js (IBM, Stongloop, 2023).

Express.js is een populair framework wat gebruik maakt van MVC-architectuur met een actieve community (PTY Ltd, 2023). De architectuur helpt bij het structureren en organiseren van de applicatie. Express.js heeft daarnaast ook ingebouwde routing, middleware ondersteuning, en heeft heel weinig overhead waardoor het een zeer snelle prestatie heeft.

Gezien de performance is voor het opslaan van productgegevens gekozen voor een NoSQL database. Voor de POC’s wordt omtrent tijd voor alle gegevens gebruik gemaakt van de NoSQL MongoDB database.

Het verzenden van e-mails alsmede het afhandelen van betalingen wordt afgehandeld via externe systemen die hierop zijn gespecialiseerd. Hierdoor wordt een stuk abstractie toegepast en wordt de kans op fouten alsmede de tijd tot aan een MVP(minimum viable product) verkleind.

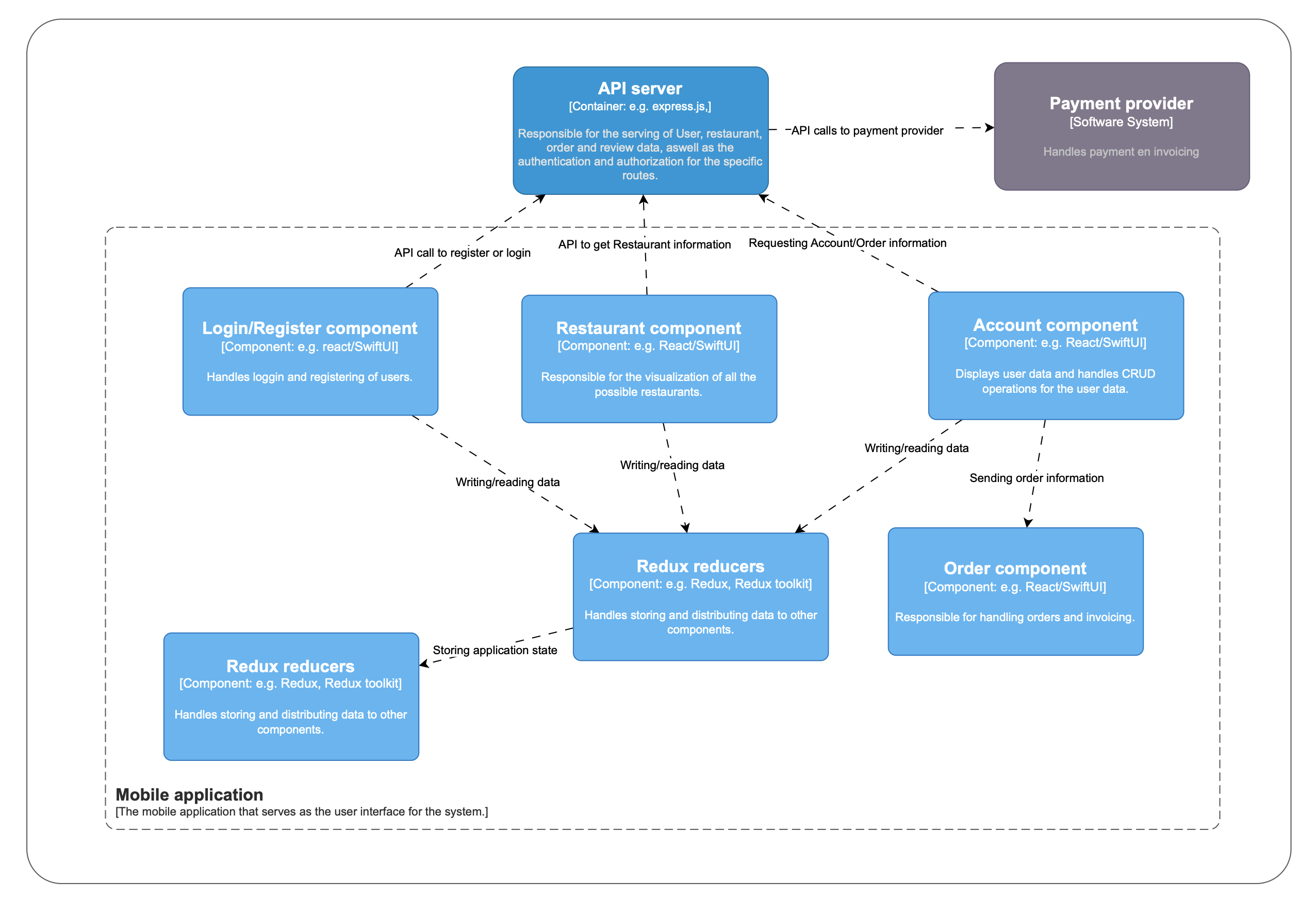


Figuur : C4 model niveau 2 System level

De mobiele front-end applicatie heeft een drietal componenten die communiceren met de API server. Deze opsplitsing zorgt ervoor dat de taken worden verdeeld en dat het mogelijk is om bijvoorbeeld restaurant gegevens in te zien zonder in te loggen.

Vervolgens wordt middels het statemanagementsysteem Redux de afhandeling van de gegevens verwerkt. Redux is een veel gebruikte manier van state management die vooral goed past binnen de applicatie. Immers moet er redelijk veel state op applicatie niveau worden bijgehouden.

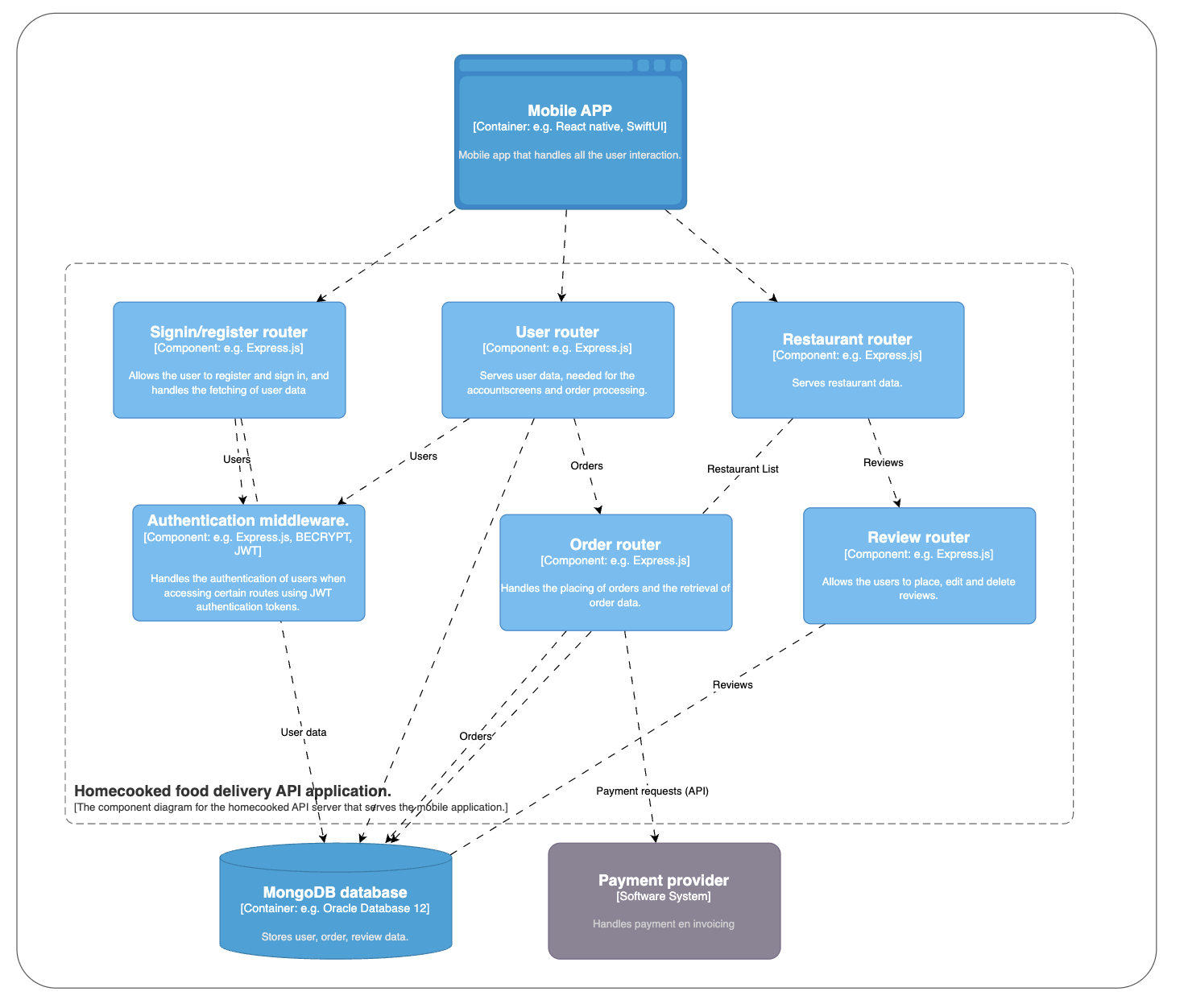
De overige componenten zijn verdeeld naar taken die zij afhandelen. Hierdoor wordt als het ware seperation of concerns toegepast. Hoewel dit voor een front-end niet noodzakelijk is, aangezien er niet veel business logica zit, vereenvoudigt het wel de applicatie.



Figuur C4 model niveau 3 component level frontend

In het component level diagram van de API server zijn wederom de verschillende communicatie lijnen opgesplitst en corresponderen zij met de zelfde componenten die bij de front-end worden gebruikt. Dit is gedaan omwille de eenvoud en de splitsing van bevoegdheden en taken.

Voor de authenticatie en autorisatie is gebruik gemaakt van een middleware die aan de verschillende routes kan worden gehangen. De wachtwoorden van gebruikers worden middels Bcrypt encryptie versleuteld opengeslagen. Waarbij een Salt aan de wachtwoorden wordt toegevoegd voor het minimaliseren van bijvoorbeeld Rainbow Table aanvallen (Wikipedia, sd).



Figuur C4 model niveau 3 component level API server

# Bibliografie

IBM, Stongloop. (2023). *Frameworks built on Express*. Opgehaald van Expressjs.com: https://expressjs.com/en/resources/frameworks.html

IBM, Strongloop. (2023). *Frameworks*. Opgehaald van Express.js: https://expressjs.com/en/resources/frameworks.html

PTY Ltd. (2023). *Express Usage Statistics*. Opgehaald van Builtwith: https://trends.builtwith.com/framework/Express

Wikipedia. (sd). *Rainbow Table*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Rainbow\_table